Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings

Zafar Shahid, William Puech

Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings

Zafar Shahid and William Puech

***Abstract*—This paper presents one of the first methods allowing the protection of the newly emerging video codec HEVC (High Efficiency Video Coding). Visual protection is achieved through selective encryption (SE) of HEVC-CABAC *binstrings* in a format compliant manner. The SE approach developed for HEVC is different from that of H.264/AVC in several aspects. Truncated rice code is introduced for binarization of quantized transform coefficients (QTCs) instead of truncated unary code. The encryption space (ES) of *binstrings* of truncated rice codes is not always dyadic and cannot be represented by an integer number of bits. Hence they cannot be concatenated together to create plaintext for the CFB (Cipher Feedback) mode of AES, which is a self-synchronizing stream cipher for so-called AES-CFB. Another challenge for SE in HEVC concerns the introduction of context, which is adaptive to QTC. This work presents a thorough investigation of HEVC-CABAC from an encryption standpoint. An algorithm is devised for conversion of non-dyadic ES to dyadic, which can be concatenated to form plaintext for AES-CFB. For selectively encrypted *binstrings*, the context of truncated rice code for binarization of future syntax elements is guaranteed to remain unchanged. Hence the encrypted bitstream is format-compliant and has exactly the same bit-rate. The proposed technique requires very little processing power and is ideal for playback on hand held devices. The proposed scheme is acceptable for DRM of a wide range of applications, since it protects the contour and motion information, along with texture. Several benchmark video sequences of different resolutions and diverse contents were used for experimental evaluation of the proposed algorithm. A detailed security analysis of the proposed scheme verified the validity of the proposed encryption scheme for content protection in a wide range of applications.**

***Index Terms*—AES-CFB, CABAC, HEVC, non-dyadic encryption space, selective encryption, truncated rice code.**

I. Introduction

HEVC [8] (High Efficiency Video Coding) is the emerging video coding standard of ITU-T and ISO/IEC. HEVC achieves visual quality similar to its previous H.264/AVC High Profile, with around 30% bit-rate reduction for the *low delay* mode (I frame followed by a number of P frames) [25]. This mode is suitable for real-time applications like video-conferencing and on-line gaming. On the other hand, *random access* mode contains I, P and B frames and any part of the frame can be accessed randomly. This mode offers better compression and is suitable for storage applications and on-line movies like Netflix and Hulu. For this mode, HEVC offers around 20% bit-rate reduction on average, but with lower complexity than the H.264/AVC Baseline Profile [25].

With the inundation of digital content which can be copied and modified easily, concerns about the protection of digital content has been raised and it is relevant to analyze HEVC regarding its protection and authentication. Selective encryption (SE) is used to restrict access of video content to only authenticated users, wherein a small part of the bit-stream is encrypted with minimal resource overhead and sufficient protection is provided for most applications. In this work, we present a selective encryption method for CABAC *binstrings* of HEVC.

Arithmetic coding is very sensitive to change in a single bit and affects the format compliance of the whole bitstream. Format compliant SE is performed on CABAC-*binstrings* (instead of CABAC bitstreams), as explained in detail in previous work of Shahid *et al.* [21], [22].

In comparison to SE-H.264/AVC, SE-HEVC poses two more challenges. The first is the non-dyadic encryption space (ES). For SE, ES of truncated rice binstrings in HEVC is not always dyadic. Hence we do not have an integer number of bits to prepare the plaintext. This challenge has been successfully addressed by conversion of non-dyadic ES to dyadic ES. The second challenge concerns the context modeling of truncated rice code. For truncated rice code, which is introduced for the first time in HEVC-CABAC, the context must remain unchanged during the encryption step for the sake of format compliance. In this paper, we have successfully addressed both of these challenges.

The rest of the paper is organized as follows. In Section II, overview of HEVC and its CABAC entropy engine is presented. This is followed by a description of recent work on selective encryption in Section III. We explain the proposed algorithm in Section IV. Section V contains an experimental evaluation and security analysis. There is also a comparison of the proposed technique with recent techniques. In Section VI, we present the concluding remarks about the proposed scheme.

II. HEVC and its CABAC Entropy Engine

HEVC has better compression efficiency as compared to H.264/AVC due to more sophisticated compression tools. One of the main ways to improve the compression rate of high-resolution videos HD is to introduce larger block structures with flexible sub-partitioning [28]. In HEVC, MB is replaced by a coding unit (CU). CU divides a video frame into a number of rectangular regions. A CU contains one or several prediction

Manuscript received September 25, 2012; revised January 23, 2013; accepted June 02, 2013. Date of publication September 06, 2013; date of current version December 12, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ton Kalker.

The authors are with the Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II, 34392 Montpellier Cedex 05, France (e-mail: zafar.shahid@lirmm.fr; william.puech@lirmm.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2013.2281029

units (PUs) and transform units (TUs). The basic partition geometry of all of these elements is encoded by using quad tree coding.

Unlike H.264/AVC, the motion vector in HEVC is predicted from four spatial neighbors (top, top-right, left, left-bottom) and one co-located MV. Consequently, an encoder has to keep MVs of a co-located reference picture in order to find a correct MV predictor. This makes the design more complex. While in H.264/AVC, only in temporal skip mode, an encoder keeps co-located MVs. In spatial skip mode, the decoder keeps only a flag indicating whether the co-located MV is bigger or not. In I frame, spatial prediction is performed from samples of already decoded adjacent PUs, where the different modes are DC (flat average), horizontal, vertical, plane (amplitude surface), bilinear, or one of up to 28 angular directions (number depending on the block size). In [13], Lainema and Ugur proposed a directional intra prediction method for HEVC. They demonstrated that the proposed method outperforms the H.264/AVC intra prediction approach on average by 4.8%. While the coding efficiency gains become more significant and exceed 10% for sequences with dominant directional structures. In the entropy coding stage, HEVC introduced the concept of entropy slices. Entropy slices do not depend on information outside of the entropy slice and can be decoded independently. This enables parallelization of the entire entropy decoding loop, including context adaptation and bin coding.

Context adaptive binary arithmetic coding (CABAC) is the only entropy coding technique, which is supported in HEVC. HEVC-CABAC consists of three stages: 1) binarization, 2) context modeling, 3) binary arithmetic coding (BAC). In the binarization step, non-binary syntax elements are converted to binary form in so-called *binstrings* which are more amenable to compression by BAC. Binary representation for a non-binary syntax element is done in such a way that it is close to the minimum redundancy code. An individual bit in a *binstring* is called a bin. The bins are coded using either regular-BAC or bypass-BAC. In regular-BAC mode, bins are passed to the context modeling step followed by BAC. On the other hand, BAC is performed using a fixed context in bypass-BAC mode.

In HEVC-CABAC, there are five basic code trees for binarization, namely the *unary* code, *truncated unary* code, *truncated rice* code with context p (TRp), k^{th} *order Exp-Golomb* code (EGk) and *fixed length* code.

1) For an unsigned integer value $x \geq 0$, the *unary* code consists of x 1's plus a terminating 0 bit.
2) The *truncated unary* code (TU) is only defined for x with $0 \leq x \leq s$. For $x < s$, the code is given by the *unary* code, whereas for $x = s$ the terminating '0' bit is neglected.
3) The *truncated rice* code with context p (TRp) is introduced in HEVC for the first time. A TRp binarization is a concatenation of quotient (q) and remainder (r) for a context p. For an unsigned integer value $x \geq 0$, the quotient q is given by $q = \lfloor x/p \rfloor$ and the remainder r is given by $r = x - qp$. For $p = 0$ the TR0 binarization is exactly the TU binarization.
4) The EGk code is also a concatenation of prefix and suffix parts. For a given unsigned integer value $x > 0$, the prefix part of the EGk *binstring* consists of a unary code corresponding to the length $l(x) = [\log_2(x/2k + 1)]$. The EGk suffix part is computed as the binary representation of $x + 2^k(1 - 2^{l(x)})$ using $k + l(x)$ significant bits. Consequently, for EGk binarization, the code length is $2l(x)+k+1$. When $k = 0$, the code length is $2l(x) + 1$.
5) The *fixed length* code is applied to syntax elements with a nearly uniform distribution or to syntax elements, for which each bit in the *fixed length* code *binstring* represents a specific coding decision e.g., *coded block flag*.

In HEVC-CABAC, the QTC syntax element is binarized by concatenation of the basic code trees. Binarization of QTC is done by REG0 (concatenation of TRp and EG0). It differs from H.264/AVC wherein QTC is binarized using UEG0 (concatenation of TU and EG0). For QTCs, binarization and BAC steps are applied to the syntax element $coeff\_abs\_value\_minus\_3 = abs\_level - 3$, since QTCs of zero magnitude are encoded using a significant map. For motion vector differences (MVD), *binstrings* are constructed by EG1.

III. Recent Work on Selective Encryption (SE) of Video Content

With the increase in digital video content, SE has attracted the attention of the research community for protection of copyright content. Since video content is huge in size and stored in compressed form, many researchers have proposed to perform encryption during the different compression steps, *e.g.,* pixel domain, transform domain, quantized transform domain or bitstream domain.

In [9], Jiang *et al.* propose to encrypt all intra prediction modes (IPMs) by chaotic pseudo-random sequence. It is followed by their scrambling by circulating sequences controlled by keys and gives a key distribution and synchronization scheme. The proposed scheme presents a good level of security but with a slight change in bit-rate. In [17], SE of H.264/AVC is carried out in some fields like intra-prediction mode, residual data, inter-prediction mode and motion vector difference (MVD). A scheme for commutative encryption and watermarking of H.264/AVC is presented in [16]. Here SE of some MB header fields is combined with watermarking DCT coefficient magnitude. This scheme presents a watermarking solution in an encrypted domain without exposing the video content. The drawback of the techniques proposed in [16], [17] is that they are not format compliant. Wang *et al.* [26] presented the partial encryption scheme on the codewords of 4×4 and 16×16 intra prediction mode (IPM), EGk code for MVD and level_suffix by using an RC4 stream cipher. Li *et al.* [14] devised a selective encryption technique for H.264/SVC on both entropy coders. The scheme encrypts IPM with signs of textures for base layers by using the stream cipher Leak Extraction (LEX) algorithm. Park and Shin [19] proposed a partial encryption of H.264/SVC scheme where the IPM, MVD and texture sign bits are encrypted. However, the IPM encryption proposed in [19], [26] affect the compression efficiency by negatively changing the video statistics.

Yeung *et al.* proposed perceptual video encryption at the transform stage by selecting one out of multiple unitary transforms [31]. The unitary transforms were significantly different from the discrete cosine transform (DCT) or discrete sine transform (DST), and the resulting coding efficiency is very close to

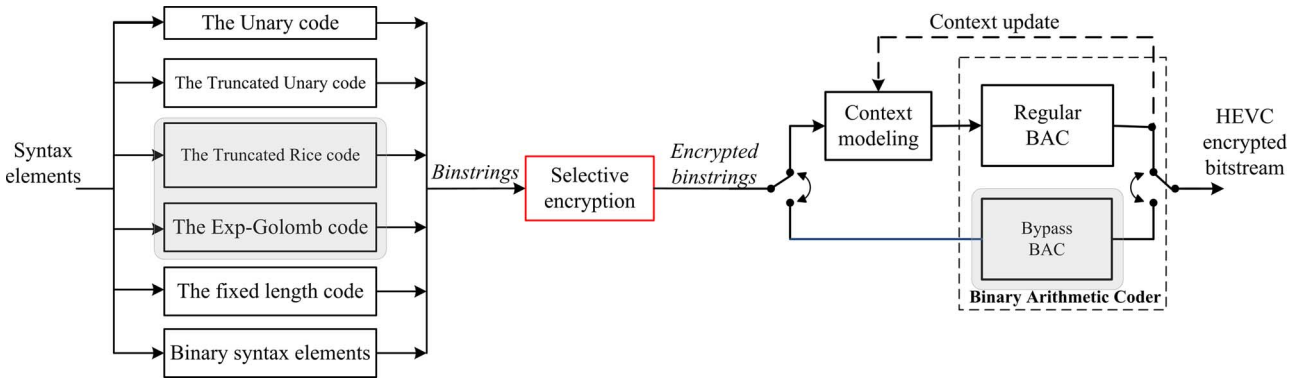


Fig. 1. Block diagram for selective encryption of *binstrings* of HEVC-CABAC. Gray-colored binarization codes are encryptable.

DCT. In [32], Yeung *et al.* extended their SE based on a unitary transform to the transforms of size 8×8 for high profiles of H.264 and MPEG4. The main drawback of a transform-coding based scheme is that it requires modification in the transform module of codec, which is very unlikely for hardware codec chips and even for DSP codecs. Moreover, it is a challenge to keep all transforms in the instruction cache, especially for embedded devices. Carrillo *et al.* proposed a partial encryption technique for H.264/AVC wherein they do permutations of pixels of MBs which are in the region of interest (ROI) [4]. The drawback of this scheme is that the bit-rate increases as the ROI size increases. This is due to a change in the ROI statistics as it is no longer a slow varying region, which is the basic assumption for video signals.

Huffman table based selective encryption has been studied in the literature in [30]. It encrypts by using different Huffman tables for different input symbols. The tables, as well as the order in which they are used, are kept secret. This technique is vulnerable to known plaintext attacks as explained in [10]. Key-based interval splitting of arithmetic coding (KSAC) has used an approach [11] wherein intervals are partitioned in each arithmetic coding iteration. A secret key is used to decide how the interval will be partitioned. The number of subintervals in which an interval is divided should be kept small as it increases the bitstream bit-rate. Randomized arithmetic coding [7] is aimed at arithmetic coding but instead of partitioning of intervals like in KSAC, a secret key is used to scramble the order of intervals. The drawback of these techniques is that the encrypted bitstream is not format compliant. Moreover, these techniques require lot of processing power.

In [5], Dubois *et al.* proposed format compliant reduced selective encryption for H.264/AVC, wherein the percentage of encrypted bits in the H.264/AVC bitstream was reduced while keeping the minimum level of visual quality. The video content was pre-analyzed to determine whether the quality had already deteriorated enough because of spatial and temporal prediction, or whether they should be selectively encrypted.

CABAC based format compliant SE has been a challenging topic in the recent past because arithmetic coding is very sensitive to errors and a change of a single bit makes the whole bitstream non-format compliant. Lee *et al.* proposed scrambling based encryption for CABAC of H.264/AVC. They proposed to adjust initialization tables of CABAC thus making the bitstream non-format compliant. Moreover, this resulted in a bit-rate overhead because of the change in the context model. In our previous work [21], [22], the novel idea of performing SE on CABAC *binstrings* instead of bitstreams for the sake of format compliance was presented. In this work, it is explained that format compliance can only be achieved if: 1) *binstrings* are selectively encrypted instead of bitstreams, 2) binstrings which we want to encrypt must be using bypass-BAC wherein a fixed context is used. Recently, Asghar *et al.* also presented an approach for the protection of scalable H.264/AVC while encrypting CABAC *binstrings* [2].

In this paper, we present a technique for HEVC content protection by selective encryption of HEVC-CABAC *binstrings*, while fulfilling real-time constraints, by transforming non-dyadic ES to dyadic and by extending SE to the header information. In Section IV, we describe our proposed approach to simultaneously apply format compliant SE and HEVC compression on video sequences in real time.

IV. The Proposed Selective Encryption Schemes

Selectively encrypted HEVC bitstream will be format compliant and will fulfill real-time constraints provided the following conditions are fulfilled:

- **Same bit-rate:** the encrypted *binstring* must has the same length as the original *binstring*.
- **Format compliance:** the encrypted *binstring* must be valid and decodable by an entropy decoder.
- **Dyadic encryption space:** ES is defined as the number of valid values for a binstring. The candidate *binstrings* should have a dyadic ES, which can be represented by an integer number of bits. This is mandatory for real-time selective encryption using AES-CFB, wherein a number of *binstrings* are concatenated to prepare a plaintext for AES-CFB.

Format compliant SE is performed on a subset of *binstrings*, which fulfills the real-time constraints as stated above. *Binstrings* are then coded by BAC as shown in Fig. 1.

Among all five binarization techniques, the *truncated rice* code with context p (TRp) and k^{th} *order Exp-Golomb* (EGk) code meet the conditions of SE, as shown in Fig. 1. The *unary* and *truncated unary* codes have different code lengths for each input value. They do not fulfill the first condition and their encryption will change the bitstream bit-rate. In *fixed length* code,

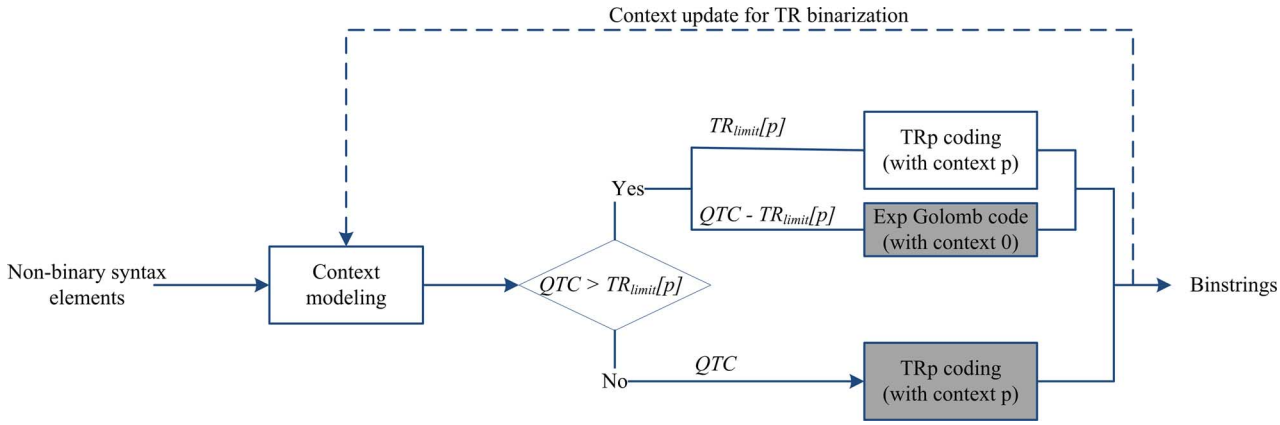


Fig. 2. Binarization of QTCs using REG0 (concatenation of TRp and EG0). Context modeling is also performed for truncated rice code. Format compliant SE is performed on gray-colored blocks.

different bits indicate different information regarding the header and is not viable for format compliant SE. Suffixes of EGk and of TRp can be encrypted while keeping the bit-rate unchanged. TRp and EG0 are used for binarization of QTC. For MVD, EG1 binarization is used. Many of EG1 *binstrings* have the same length and hence the first and second conditions are fulfilled. Since MVDs are part of the CU (coding unit) header and are used for prediction of future motion vectors, it should be guaranteed that the encrypted motion vector will lie in the valid range.

A. Encryption Space in an HEVC Bitstream

The encryption space for HEVC is constituted by the sign bit of QTCs, suffix of TRp code, suffix of EG0 code, sign of MVDs and suffix of EG1 code. Context p must remain unchanged for both the TRp code and BAC step. This is in contrast to SE-H.264/AVC wherein BAC was the only component using context modeling. From the CABAC entropy engine standpoint, we can encrypt only *binstrings* which use bypass-BAC mode and fixed contexts, as highlighted in Fig. 1. Contexts for *binstrings* coded by regular-BAC are adaptive and their encryption makes the bitstream non-format compliant because of context mismatches on the encoder and decoder side. All *binstrings* which are coded by bypass-BAC are not encryptable. Rather, only *binstrings* which guarantee same bit-rate and same TRp context can be encrypted.

For binarization of QTCs, UEG0 code is used in H.264/AVC. It is replaced by REG0 (concatenation of TRp and EG0) in HEVC [18]. The main reason for using TRp code is to increase the number of bypass bins by coding QTCs up to a specific maximum value with TRp. Moreover, EGk code is optimally fit for distribution of H.264/AVC residuals, which is geometrical. While the distribution of HEVC residuals is such that they are better compressed by TRp codes.

The QTCs are denoted by *levels* in this section. Binarization of *levels* is performed using **only** TRp code up to a threshold, as shown in Fig. 2. The threshold depends on context p ([8, Table 9.35]) and is given by [12]:

$$TR_{limit}[p] = \{7,\ 14,\ 26,\ 46,\ 78\}. \tag{1}$$

1) Scenario I: $level \leq TR_{limit}[p]$: If $|level|$ is smaller than $TR_{limit}[p]$ for context p, it will be binarized using only TRp code, whose suffix will be encrypted as shown in Fig. 2. Context p for TRp code can increment by 1 at a time and is adaptive to $|level|$ in the following manner:

Algorithm 1 Selection of context p is adaptive to $|level|$.

1: **if** $|level| > 3 * (1 \ll p_{curr})$ **then**

2: $p_{next} \leftarrow \min(p_{curr} + 1, 4)$

3: **end if**

For example, if $level$ is 2 and the current context p_{curr} is 0, then p_{next} will remain unchanged. If the level is modified from 2 to 3 during the encryption step, then p_{next} will be 1. Context p_{next} must remain unchanged during encryption of QTCs otherwise the encrypted bitstream will not be format compliant. In a TRp code with context p, we have p encryptable bits except for two limitations:

- $TR0$ (*truncated rice* with context 0) code is the same as *truncated unary* code and each *binstring* of $TR0$ has a different length and is not encryptable because of violation of the bit-rate constraint.
- The length of the last equal-length group of TRp codes is the same whether the EG0 code is appended or not. So if we encrypt the *binstrings* of this group, it may make the bitstream non-compliant. To fulfill this constraint, the last *binstring* is excluded from the set of equal-length *binstrings*, thus making the encryption space **non-dyadic**.

The selection of encryptable bits of the TRp suffix is illustrated in Algorithm 2. Note that the encryption space for a *binstring* is selected in such a way that it does not affect context p and its length.

Algorithm 2 Encryption space in a TRp suffix which fulfills format compliant constraints for selective encryption of HEVC.

Require: $p > 0$

Require: $|level| < TR_{limit}[p_{curr}]$

1: $TR_{limit}[p_{curr}] = \{7,\ 14,\ 26,\ 46,\ 78\}$

2: $limit\_rice\_para[p_{curr}] = \{3,\ 5,\ 12,\ 24, \infty\}$

3: $default\_ES \leftarrow 2^p$

4: **if** $|level| \leq limit\_rice\_para[p_{curr}]$ **then**

5: $upper\_bound = limit\_rice\_para[p_{curr}]$

6: $lower\_bound = (prefix \ll p)$

7: **else**

8: $upper\_bound = TR_{limit}[p_{curr}]$

9: $lower\_bound = limit\_rice\_para[p_{curr}] + 1$

10: **end if**

11: **if** $((prefix \ll p) + default\_ES) \leq upper\_bound$ **then**

12: $upper\_bound = (prefix \ll p) + default\_ES - 1$

13: **end if**

14: **if** $(prefix \ll p) > lower\_bound$ **then**

15: $lower\_bound = (prefix \ll p)$

16: **end if**

17: $ES \leftarrow ((upper\_bound - lower\_bound) + 1)$

2) Scenario II: $level > TR_{limit}[p]$: If $|level|$ is higher than a $TR_{limit}[p_{curr}]$ limit, EG0 code is used for the binarization of $level - TR_{limit}[p_{curr}]$. In this case, the TRp code is fixed (all bits are '1') and is not encryptable. The suffix of EG0 will be encrypted in this case, while guaranteeing the same *binstring* length and same future context p_{next} for the TRp code, as shown in Fig. 2.

The sign and suffix of the MVD syntax element are also candidates for SE. HEVC uses EG1 code for binarization of MVDs. Encryption of MVD suffixes is crucial for protection of contours and motion information but in this case we cannot guarantee the format compliance.

B. Conversion of Non-Dyadic ES to Dyadic ES

To prepare plaintext for AES-CFB, we require suffixes having dyadic (*i.e.,* power of 2, which can be constituted by integer number of bits) ES. In contrast to H.264/AVC, wherein ES was composed of EG0 suffixes and was always dyadic, it is a common scenario with TRp suffixes in HEVC to have non-dyadic ES. This is due to the fact that the threshold for the context p of TRp binary codes lies at non-dyadic boundaries, as given in Algorithm. 1. Real-time SE-HEVC using AES-CFB is only possible if non-dyadic ES is transformed to dyadic ES. This is achieved by decomposition of non-dyadic ES into small dyadic ES as explained in the following steps:

1) Let L be the non-dyadic ES which is to be decomposed into n smaller dyadic ES $l_{dyadic}[n]$.
2) The first dyadic ES $l_{dyadic}[0]$ consists of the first 2^k suffixes $(k|\max(k) \wedge 2^k \leq L)$.
3) This process is repeated on the remaining ES $(L - 2^k)$ recursively to decompose L into dyadic ES.
4) Dyadic ES $l_{dyadic}[i]$ that contains the *suffix* to be encrypted, is the available dyadic ES for that suffix.

For example, a non-dyadic ES with ES $= 14$ will be decomposed into three dyadic ES with ES $= 8, 4$ and 2 respectively. If the suffix is on index 5, its ES will be the first dyadic ES $\{0, 1, 2, \ldots 7\}$ with $ES = 8$. Similarly if the suffix to be encrypted has an index value of 9 or 13, their ESs will be $\{8, 9, .., 11\}$ and $\{12, 13\}$, with $ES = 4$ and $ES = 2$ respectively. It is important to note that the last dyadic ES $l_{dyadic}[n-1]$ may have $ES = 1$. In that case, the suffix on the final index in the ES will not be encrypted. For example, for ES $= 15$, the dyadic ES will be of sizes 8, 4, 2 and 1. In this case, the last suffix cannot be encrypted because it has dyadic ES with $ES = 1$.

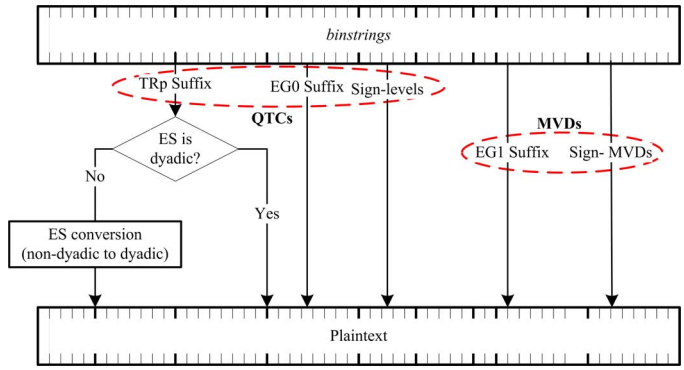


Fig. 3. Preparation of plaintext for AES-CFB for SE of HEVC-CABAC.

Fig. 3 illustrates the preparation of plaintext for the proposed method for SE of HEVC. For QTCs, the sign and suffix of either the TRp or EG0 *binstring* is a candidate for SE. While for motion information, the sign and suffix of EG1 binstrings are encrypted.

C. Selective Encryption of HEVC-CABAC Binstrings

HEVC has introduced the concept of entropy slices. Context models are reset at the start of every entropy slice. Moreover, entropy slices restrict the neighborhood definition. SE of HEVC is performed on each entropy slice independently. Let us consider $Y_i = X_i \oplus E_k(Y_{i-1})$ as the notation for the encryption of a n bit block X_i, using the secret key k with the AES-CFB. We have chosen to use this mode in order to keep the original compression rate. Indeed, with the CFB mode for each block, the size of the encrypted data Y_i can be exactly the same as the size of the plaintext X_i. In this mode, the code from the previously encrypted block is used to encrypt the current one.

SE is performed on *binstrings* before compression by BAC. Non-binary syntax elements are transformed into *binstrings* through binarization and at the same time we fill X_i with encryptable bits until either the vector X_i is completely filled or the slice boundary is reached. Let $L(X_i)$ be the length up to which vector X_i is filled. In case of a slice boundary, if $L(X_i) < C$, we apply a padding function $p(j) = 0$, where $j \in \{L(X_i)+1, \ldots, C\}$, to fill in vector X_i with zeros up to C bits. Note that the main difference with respect to H.264/HVC is that we have to prepare the plaintext while getting encryptable bits from one entropy slice and we have to pad it with zeros if the end of the entropy slice is reached.

In AES-CFB, the previous encrypted block Y_{i-1} is used as the input of the AES algorithm in order to create Z_i. Then the current plaintext X_i is XORed with Z_i in order to generate

TABLE I
THE SET OF BENCHMARK VIDEO SEQUENCES USED TO EVALUATE THE PERFORMANCE OF THE PROPOSED SE-HEVC TECHNIQUE [3]

Class	Resolution	Framerate	Videos
A	2560×1600	30	Traffic (S01), PeopleOnStreet (S02)
B1	1920×1080	24	Kimono (S03), ParkScene (S04)
B2	1920×1080	50-60	Cactus (S05), BasketballDrive (S06),BQTerrace (S07)
C	832×480	30-60	BasketballDrill (S08), BQMall (S09), PartyScene (S10), RaceHorses (S11)
D	416×240	30-60	BasketballPass (S12), BQSquare (S13), BlowingBubbles (S14), RaceHorses (S15)
E	1280×720	60	Vidyo1 (S16), Vidyo2 (S17), Vidyo3 (S18)

the encrypted text Y_i. Moreover, generation of the encrypted stream Z_i depends on the previous encrypted block Y_{i-1}. Consequently, if two plaintexts are identical $X_i = X_j$ in CFB mode, then the two corresponding encrypted blocks are always different, $Y_i \neq Y_j$.

The decryption process for AES-CFB works in the same fashion except that the input is encrypted and the output will be the plaintext. The decoded plaintext vector is split into suffixes in order to substitute the encrypted *binstrings* with the original ones. The bitstream is then further decoded to get the decrypted video content using standard HEVC decode steps.

V. EXPERIMENTAL RESULTS

In this section, different aspects of the proposed schemes for SE-HEVC are analyzed. For simulation purposes, reference implementation of HEVC HM 8.0[1] was used. The set of benchmark video sequences along with their resolutions is given in Table I. This is the same set of sequences that is being used in the HEVC standardization process [3].

We simultaneously applied our proposed SE scheme and HEVC compression as described in Section IV, on all the benchmark video sequences for *low delay* and *random access* mode. The *intra period* is 10 for both modes for a sequence of 50 frames. An I frame is preceded by 9 P frames in *low delay* mode. In *random access* mode, two B frames are inserted between consecutive P frames. First, an analysis of available ES is presented in Section V-A. It is followed by a PSNR analysis of the proposed scheme in Section V-B. Section V-C demonstrates the processing efficiency of the proposed technique and infers that it is suitable for smartphones, tablets and other battery operated devices. Security analysis of SE-HEVC is discussed in Section V-D, wherein security related features like entropy, standard deviation and histogram are presented to confirm that the proposed scheme offers sufficient security. A comparative analysis of the proposed scheme for SE-HEVC with recent work is presented in Section V-E.

A. Encryption Space for SE-HEVC

Encryption space (ES) is defined as the percentage of encryptable bits in a video bitstream. It varies from one sequence to another based on the video content. In Table II, ES for different benchmark video sequences is presented for the QP value 18. Note that ES varies from 16.96% to 20.08% based on the video content. Video sequences with either static background or translational movement (*e.g., Vidyo1*) have less ES. *BQMall* also has less ES despite its complex background. This is due to the fact that the background is static and coded by *skip* blocks and translational motion in the foreground is very efficiently predicted. On the other hand, video sequences which have complex motion and moving background have high ES. For example, *RaceHorses* contains walking horses, camera movement and high-texture grass in the background and has higher ES. The ES for EG1 suffixes of MVD syntax elements depends on the complexity of the motion in the video sequence. For a simple translational motion, which can be easily estimated, ES for EG1 suffixes of an MVD syntax element will be less, while it will be higher for video sequences with complex motion.

Table III provides an analysis of the effect of QP on the ES for the *kimono* video sequence. Note that the ES *slightly* decreases with an increase in the QP value. In a video bitstream, the data part decreases with an increase in QP, while the proportion of video header (NAL header, slice header, CU/PU/TU headers etc.) increases with an increase in the QP value. For variations in the QP range from 18 to 42, the change in ES for SE-HEVC in *low delay* mode is from 19.93% to 16.56%, as shown in Table III. This is in contrast to the SE-CABAC of H.264/AVC wherein ES decreases from 19.97% to 9.46% ([22, Table II]) for the same QP range. This is because of the replacement of truncated unary (TU) coding with truncated rice (TR) coding for QTC binarization.

Tables II and III also provide a comparison of *low delay* (I, P frames) and *random access* (I, P, B frames) modes for PSNR and ES. Generally, the PSNR of B frames is slightly less as compared to that of P frames, which results in an overall decrease in PSNR for *random access* mode. ES in *low delay* mode is also slightly higher than in *random access* mode. Since *random access* mode is more generic and contains I, P and B frames, the results will be presented only for *random access* mode from hereon for the sake of brevity.

B. Visual Protection of SE-HEVC for I, P and B Frames

To demonstrate the visual protection offered by our proposed scheme, we compressed 50 frames in *random access* mode with 2 B frames between consecutive P frames. Video frame areas that contain many details and texture will have lot of non-zero QTCs and consequently will be strongly encrypted. On the other hand, homogeneous areas in a video frame, *i.e.,* areas containing series of identical pixels, are less ciphered. The SE of MVD syntax elements is helpful for the protection of motion information in a video sequence. Table IV compares the average PSNR of all benchmark video sequences without encryption and with

[1]https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

TABLE II
ES ANALYSIS OF SE-HEVC FOR BENCHMARK VIDEO SEQUENCES AT A QP VALUE OF 18 FOR *LOW* DELAY AND *RANDOM* ACCESS MODES. ES IS LESS FOR SEQUENCES WITH EITHER STATIC BACKGROUND OR TRANSLATIONAL MOTION, WHILE IT IS HIGHER FOR COMPLEX VIDEO CONTENT

Class	Sequence	*low delay*			*random access*		
		PSNR(Y) (dB)	ES (%)	ES-Class (%)	PSNR(Y) (dB)	ES (%)	ES-Class (%)
A	Traffic	44.08	18.03	18.47	43.72	18.1	18.31
	PeopleOnStreet	43.89	18.91		42.92	18.52	
B1	ParkScene	42.38	18.36	18.43	41.98	18.52	18.67
	Kimono	43.37	18.5		43.09	18.81	
B2	BasketballDrive	42.63	18.22	19.26	41.68	17.93	18.75
	BQTerrace	43.4	19.95		41.83	18.99	
	Cactus	41.92	19.6		40.78	19.32	
C	BasketballDrill	43.6	17.3	18.52	43.42	17.3	18.27
	BQMall	42.22	17.38		41.86	17.5	
	PartyScene	43.05	19.32		41.94	18.96	
	RaceHorses	43.15	20.08		41.87	19.33	
D	BasketballPass	44.65	17.88	18.48	44.7	17.48	18.11
	BQSquare	42.41	18.94		41.71	18.01	
	BlowingBubbles	41.91	18.44		41.27	18.4	
	RaceHorses	42.94	18.66		42.17	18.55	
E	Vidyo1	45.34	16.96	17.54	45.24	17.02	17.48
	Vidyo3	44.86	18.1		44.61	17.88	
	Vidyo4	45.35	17.57		44.89	17.53	
	Average	43.39	18.45		42.76	18.23	

TABLE III
ES ANALYSIS OF THE PROPOSED SCHEME OVER THE WHOLE RANGE OF QP VALUES FOR THE *KIMONO* VIDEO SEQUENCE FOR *LOW* DELAY AND *RANDOM* ACCESS MODE

QP	*low delay* ES (%)	*random access* ES %)
12	19.93	19.81
18	18.50	18.81
24	18.34	18.75
30	18.08	18.43
36	17.77	17.89
42	16.56	17.05

SE-HEVC for QP values 18 and 32. Average PSNR value of *luma* for all sequences is 9.67 *dB* for QP value 18, and is 10.11 *dB* for QP value 32 for SE-HEVC. This confirms that this algorithm works well for various combinations of motion, texture and objects for I, P and B frames whatever the QP value. Moreover, average PSNR values of U and V are 15.82 *dB* and 17.23 *dB* respectively for QP value 18 (and average PSNR values of U and V are 15.82 *dB* and 17.23 *dB* respectively for QP value 32), which are lower as compared to SE-CABAC of H.264/AVC values of 21.90 *dB* and 23.50 *dB* ([22, Table IX]) for QP value 18. Hence SE-HEVC provides better texture and color information protection in a video sequence. Fig. 4 shows the I, P and B frames which are simultaneously encoded and encrypted by SE-HEVC. Note that the SE of the MVD syntax element has helped for protecting the motion and structural information for P and B frames.

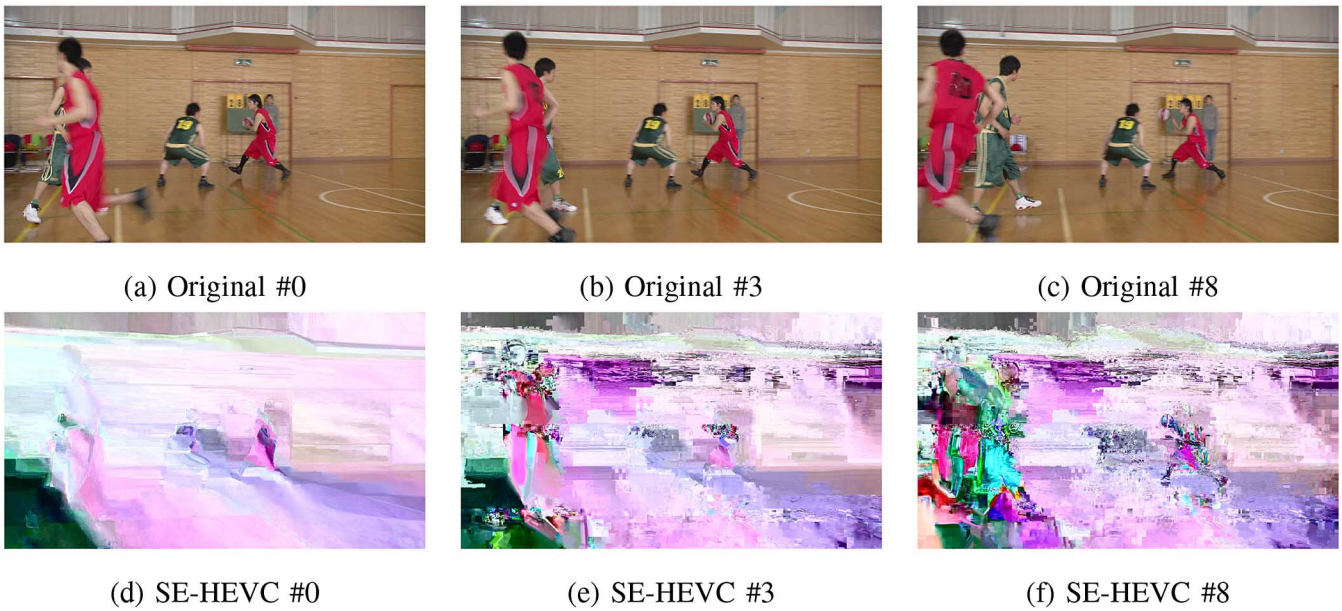
For an analysis of visual protection by SE-HEVC over the whole range of QP values, Table V compares the average PSNR of *kimono* over the whole range of QP values without encryption and with SE-HEVC. Non-zero QTCs decrease with an increase in QP and the contribution of SE-MVD in the overall protection increases with an increase in the QP value. Note that PSNR values of *luma* and *chroma* components remain in the lower range for all QP values. Fig. 5 shows original as well as SE video frame #8 of *kimono* for QP values of 18, 30 and 42. It is evident that the encrypted video frame is quite secure over all QP values in terms of texture, motion, color and luminance.

C. Computational Overhead

Computational overhead is an important factor especially for smartphones and hand-held devices which have limited processing power. It can be calculated through an analysis of the additional time required for encoding and decoding encrypted bitstreams. The computers used for this simulation had Intel Core 2 Duo T8100 processor with 3072 MB random access memory. Table VI shows the encoding and decoding time for the kimono sequence with an *intra period* of 10. The additional processing time for encoding (column 4) and decoding (column 7) is negligible as compared to the overall processing time, thus confirming the processing efficiency of the proposed scheme.

D. Security Analysis

This section presents a security analysis of the proposed SE-HEVC scheme in various aspects. The analysis includes a



(a) Original #0 (b) Original #3 (c) Original #8

(d) SE-HEVC #0 (e) SE-HEVC #3 (f) SE-HEVC #8

Fig. 4. Frames #0, #3 #8 (I, P and B respectively) of BasketballDrive sequence for SE-HEVC for QP value 18. (a) Original #0. (b) Original #3. (c) Original #8. (d) SE-HEVC #0. (e) SE-HEVC #3. (f) SE-HEVC #8.

TABLE IV
COMPARISON OF PSNR WITHOUT ENCRYPTION AND WITH THE SE OF BENCHMARK VIDEO SEQUENCES FOR *RANDOM* ACCESS VIDEO SEQUENCES (CONTAINING I, P AND B FRAMES) AT QP VALUES 18 AND 32

		QP 18						QP 32					
Class	Sequence	PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)		PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)	
		Orig.	SE	Orig.	SE	Orig.	SE	Orig.	SE	Orig.	SE	Orig.	SE
A	Traffic	44.08	9.79	43.50	14.29	45.72	18.23	36.26	9.12	38.12	15.67	40.41	18.92
	PeopleOnStreet	43.88	8.23	46.36	15.28	45.82	17.67	34.56	10.18	41.04	13.87	41.75	20.12
B1	ParkScene	42.38	10.25	43.80	15.23	45.60	14.29	34.80	8.34	38.57	8.32	40.02	16.46
	Kimono	43.37	9.55	44.80	12.78	46.83	17.27	37.68	8.40	40.35	21.49	41.82	28.01
B2	BasketballDrive	42.67	8.95	45.45	15.81	47.05	19.41	36.91	12.07	42.14	24.58	42.02	16.44
	BQTerrace	43.42	7.89	43.62	14.59	45.28	17.26	33.18	9.21	38.87	16.81	41.04	25.88
	Cactus	41.93	7.38	42.50	15.46	44.70	17.63	34.63	8.38	37.95	17.22	39.89	18.44
C	BasketballDrill	43.61	11.35	45.32	13.64	46.51	11.84	34.84	9.42	38.86	19.22	39.18	11.17
	BQMall	42.21	10.86	44.55	18.48	46.37	16.73	34.20	11.40	39.12	15.55	39.99	21.86
	PartyScene	43.04	8.99	43.86	16.13	44.55	20.23	31.11	10.14	36.19	18.65	36.48	19.99
	RaceHorses	43.17	10.64	43.95	11.22	44.68	14.48	32.39	10.62	36.90	15.11	38.05	13.60
D	BasketballPass	44.66	13.76	46.61	23.46	46.66	14.39	34.51	12.74	39.20	24.35	38.73	23.50
	BQSquare	42.39	6.72	45.53	22.20	46.22	19.19	31.49	8.20	39.57	26.31	40.07	18.53
	BlowingBubbles	41.92	12.15	43.39	12.04	45.21	22.22	32.01	12.98	36.52	14.47	38.18	22.11
	RaceHorses	42.94	8.75	44.01	16.75	44.45	16.25	36.77	10.9	36.47	13.77	31.86	15.93
E	Vidyo1	45.35	10.32	47.68	15.23	48.78	18.95	38.94	11.16	44.06	24.20	44.55	28.99
	Vidyo3	44.86	9.42	49.23	14.56	49.03	17.79	38.14	10.12	45.68	27.58	43.86	28.68
	Vidyo4	45.35	8.98	49.34	17.54	49.61	16.39	38.93	8.59	45.09	27.33	45.32	23.39
	Average	43.40	9.67	45.20	15.82	46.28	17.23	35.08	10.11	39.71	19.14	40.18	20.67

histogram comparison of original and encrypted video frames in Section V-D.1, encryption quality analysis in Section V-D.2, edge and structural protection in Section V-D.3, entropy and standard deviation analysis in Section V-D.4, key sensitivity analysis in Section V-D.5, and protection against known plain-text attack in Section V-D.6.

(a) Original (b) QP = 18 (c) QP = 30 (d) QP = 42

Fig. 5. Frame #8 (B frame) for SE-HEVC at different QP values for the *kimono* video sequence (a) original, (b-d) SE-HEVC frame at different QP values. (a) Original. (b) QP = 18. (c) QP = 30. (d) QP = 42.

TABLE V
PSNR COMPARISON WITHOUT ENCRYPTION AND WITH SELECTIVE ENCRYPTION (SE-HEVC) FOR THE *KIMONO* VIDEO SEQUENCE OVER THE WHOLE RANGE OF QP VALUES

	PSNR (Y) (dB)		PSNR (U) (dB)		PSNR (V) (dB)	
QP	Original	SE-HEVC	Original	SE-HEVC	Original	SE-HEVC
12	47.67	9.12	48.23	13.03	49.62	10.77
18	43.37	9.55	44.80	12.78	46.83	17.27
24	41.47	9.91	42.69	17.42	44.39	13.32
30	38.89	8.15	41.02	17.46	42.53	17.31
36	35.79	10.18	39.71	16.60	41.22	13.48
42	32.66	12.35	38.80	19.91	40.50	17.31

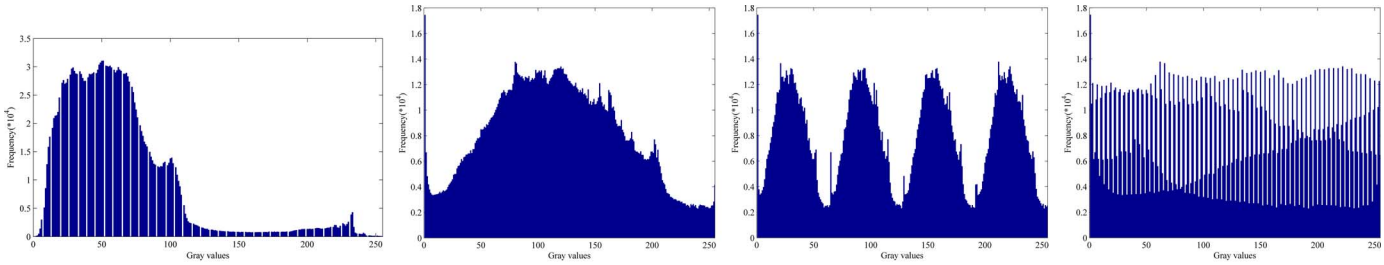
TABLE VI
ENCODING/DECODING PROCESSING TIME COMPARISON WITHOUT ENCRYPTION AND WITH SE-HEVC FOR THE *KIMONO* VIDEO SEQUENCE

	Encoding time (seconds)			Decoding time (seconds)		
Frames	Original	SE-HEVC	Diff.	Original	SE-HEVC	Diff.
10	9767	10060	293	16.15	16.59	0.44
20	19500	20159	659	30.89	32.05	1.16
30	29401	30341	940	49.15	50.70	1.55
40	38569	39610	1040	65.13	66.96	1.83
50	48009	49593	1584	79.05	82.18	2.03

1) Histogram Analysis: A histogram of a video frame gives the frequency distribution of the intensity levels. For a good encryption system, histograms of original and encrypted video frames should differ from each other. A histogram of the original *kimono* #8 video frame is shown in Fig. 6(a), while histograms of the encrypted version of the same video frame with different bit-shifts are presented in Fig. 6(b)–(d). It is evident from Fig. 6(a)–(d) that the histograms of original and encrypted frames are entirely different.

2) Encryption Quality Analysis: The encryption quality (EQ) represents the average number of changes to each gray level L [1]. Let V and V′ denote the original video frame (plainframe) and the selective encrypted video frame (cipherframe) of a video sequence of resolution $m \times n$ with L intensity levels. $V(x,y)$, $V'(x,y) \in \{0,\ldots,L-1\}$ are the gray levels of video frames V and V′ at position (x,y) $(0 \le x \le m-1, 0 \le y \le n-1)$. Let $H_L(V)$ denote the number of occurrences of each grey level L in the plainframe V. Similarly, $H_L(V')$ denotes the number of occurrences of each grey level L in cipherframe V'. The EQ can be defined as:

$$EQ = \frac{\sum_{L=0}^{255} |H_L(V') - H_L(V)|}{256}. \quad (2)$$

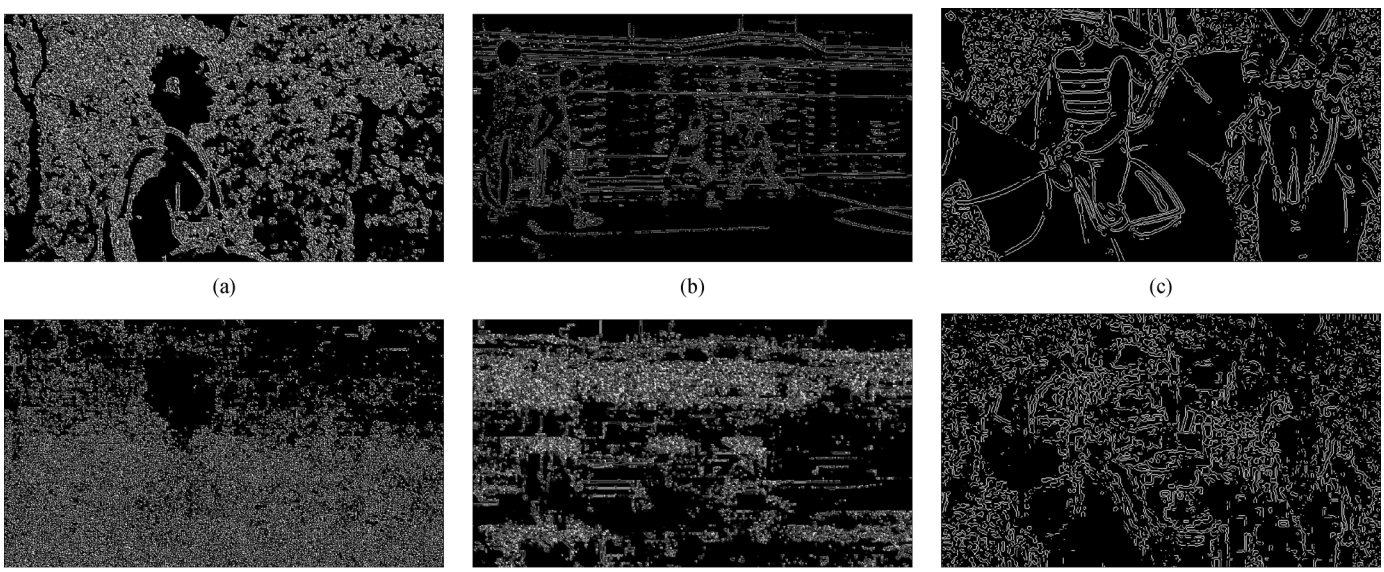


(a) (b) (c) (d)

Fig. 6. Histogram analysis of the original and encrypted versions of video frame # 8 of the *kimono* video sequence at QP value 18. (a) Original. (b) no shift. (c) 2 left shift. (d) 2 right shift.

TABLE VII
ENCRYPTION QUALITY FOR SE-HEVC OF THE *KIMONO* VIDEO SEQUENCE AT QP VALUE 18 FOR DIFFERENT BIT-SHIFTS

Left bit Shift	0	1	2	3	4	5	6	7
EQ	9019	9588	8943	8446	8472	8575	8540	8642



(a) (b) (c)

(d) (e) (f)

Fig. 7. Edges detected using the Laplace edge detector in original and encrypted video frames, wherein edges in encrypted frames are completely distorted. (a) original *kimono* #0. (b) original *basketballDrive* #0. (c) original *raceHorses* #0. (d) encrypted *kimono* #0. (e) encrypted *basketballDrive* #0. (f) encrypted *raceHorses* #0.

The encryption quality for frame #8 of *kimono* is shown for a QP value of 18 in Table VII for a different number of shifts. Note that the encryption quality remains in the higher range for all the shifts.

3) Protection of Edges and Structural Information: A good visual encryption system aims to make the visual content incomprehensible by protection of edges and contour information. The degradation in encrypted video frame can effectively be evaluated by measuring the distortion introduced at the edges. The edge differential ratio (EDR) [24] depicts a deviation in the location of edge formation contributing pixels in the original image and its encrypted video frame. The mathematical representations of EDR can be expressed as:

$$EDR = \frac{\sum_{i,j=1}^{N} \left| B(i,j) - \hat{B}(i,j) \right|}{\sum_{i,j=1}^{N} \left| B(i,j) + \hat{B}(i,j) \right|}, \tag{3}$$

where $B(i,j)$ and $\hat{B}(i,j)$ denote the pixel values in the edge detected binary version for the original and encrypted images, respectively. Fig. 7 shows the binary images after Laplace edge detection for the original and encrypted video frames. It is evident that the structural information is heavily distorted in the encrypted video frames.

Higher EDR indicates (close to '1') better protection of the structural information of a video frame, whereas a lower value (close to '0') indicates similarity between the original and encrypted video frames. EDR for frame #0 of *Kimono*, *BasketballDrive* and *RaceHorses* is 0.97, 0.92 and 0.94 respectively, which confirms that the original and encrypted video frames are entirely different. It is important to note that EDR is performed for I frames. The values would be even higher if conducted on P or B frames because of encryption of suffixes of MVD syntax elements.

4) Analysis of Entropy and Local Standard Deviation: The security of the encrypted image can be measured by consid-

(a)

(b)

Fig. 8. Key sensitivity test for *kimono* #0. The SE frame is decrypted with: (a) Original key: PSNR (YUV) = {45.78, 47.04, 48.88} dB, (b) 1-bit different key: PSNR (YUV) = {9.45, 19.11, 13.63} dB.

(a)

(b)

Fig. 9. Known plaintext attack for *kimono* #0: (a) Encrypted frame: PSNR (YUV) = {10.12, 20.65, 14.23} dB, (b) Video frame with encrypted bits set to 0: PSNR (YUV) = {11.12, 20.19, 14.69} dB.

ering the variations (local or global) in the protected image. In contrast to full encryption wherein the encrypted content has the highest entropy with high local standard deviation values, the SE-HEVC video frame is transformed to flat regions with blocking artifacts. This is generally due to variation in pixel values at the boundaries of coding, prediction or transform units (CU, PU, TU). If the probability of each gray level in a video frame is $P(\alpha_i) = 1/2^k$, then the encryption of this video frame is robust against 1*st* order statistical attacks, and thus $H(X) = \log_2(2^k) = k$ bits/pixel. In a video frame, the information redundancy r is defined as $r = k - H(X)$. For all the benchmark video sequences, the average information redundancy r for SE-HEVC sequences is 0.52, while it is 1.01 for all of the original sequences. Moreover SE-HEVC has less information redundancy than SE-CABAC of H.264/AVC ([22]) which was 0.55. For all benchmark video sequences, the mean local standard deviation of *luma* is 65.15 for SE-HEVC bitstreams, where the mean local standard deviation is less than 10 gray levels for the original benchmark sequences. Note that the local standard deviation of encrypted sequences is much higher than for the original sequences.

5) Key Sensitivity Test: To guarantee the security of a cryptosystem against brute-force attacks, the cryptosystem should be highly sensitive towards the key. So the ciphertext cannot be decrypted correctly although there is only a slight difference between the encryption or decryption keys. For this purpose, a key sensitivity test is assumed where we picked one key and then applied the proposed technique for encryption and then made a one bit change in the key and decoded the bitstream. The numerical results show that the proposed technique is highly sensitive towards the key change, that is, a different version of the encrypted video sequence is produced when the keys are changed, as shown in Fig. 8. PSNR of *luma* of decrypted frames with 1-bit different key is 9.45 dB, which lies in the same lower range as the encrypted video frames.

6) Known Plaintext Attack: In known plaintext attack, the encrypted bits are guessed based on the non-encrypted information available in the SE-HEVC bitstream. The sign bits of QTCs and MVDs are either 0 or 1 and an attacker can recover them through a brute force attack. The encryption of **TRp & EG0 suffixes for QTCs** and **EG1 suffixes for MVDs** plays a major role in making the scheme robust to this attack. Encrypted bits are substituted by a constant value in order to measure the strength of the proposed SE-HEVC method as described in [20]. Here we used frame #0 of *kimono* with QP value 18. Fig. 9 shows both encrypted and attacked video frames for SE-HEVC. For example, *luma* of the SE-HEVC video frame has PSNR of 10.12 *dB* (Fig. 9(a)), while the attacked SE-HEVC video frame has PSNR of 11.02 *dB* (Fig. 9(b)). This confirms that the proposed SE scheme is robust against such known-plaintext attacks.

E. Comparative Analysis

The proposed algorithm is pioneer work on SE-HEVC and has successfully addressed the challenges posed by its entropy engine. In this section, a comparison of our proposed SE scheme with recent work is performed to verify the effectiveness of the proposed scheme. Table VIII presents a detailed comparison between the proposed SE-HEVC and recent work. Previous methods on selective encryption of video codecs have disadvantages over compression ratio, format compliance and/or security of implemented schemes by choosing weak encryption parameters and/or weak cipher algorithms. Only residual data and sign bit encryption in [22], [23], [27], [29] are considered to provide low level perception security, and the video can be recovered by brute force attacks.

Although, the pixel domain encryption proposed by Carrillo *et al.* [4] is robust to transcoding, it has the disadvantage of being independent of the video compression system and requires complete re-encoding. Yeung *et al.* [31] proposed encryption in the transform domain. This technique has the limitation of requiring complete re-encoding of the video bitstream and thus a lot of processing power. Moreover, this technique is not compliant with respect to the respective video standard and requires alternating transforms. IPM encryption is proposed by several researchers [9], [14], [19], [26]. It changes the video statistics in a negative manner. This is because absolute values are used and thus the change in any bit of the absolute value causes bit-rate fluctuation.

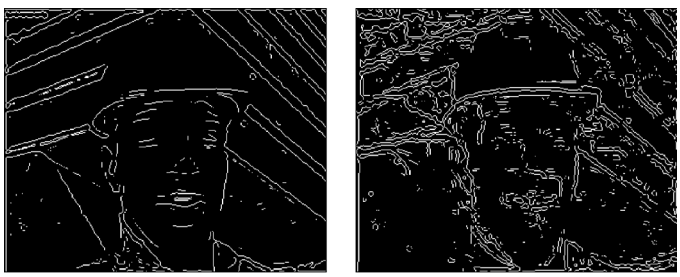
Along with its processing efficiency and format compliance, the proposed scheme has several distinct features. For example, ES is only slightly affected over the whole range of QP values. When the QP varies from 18 to 42, ES of SE-HEVC varies from from 19.93% to 16.56% (Table III). While for H.264/AVC, ES decreases from 19.97% to 9.46% ([22, Table II]) over the same QP range. Moreover, SE-HEVC has addressed one more challenge successfully, which is conversion of non-dyadic ES to dyadic ones for encryptable suffixes. Non-dyadic ES does not allow formation of a plaintext for AES-CFB from encryptable binstrings.

The motion and contour information of the video content is better protected in the proposed scheme as compared to previous methods based on residual data and sign bit encryption. Fig. 10

TABLE VIII
COMPARISON OF OUR PROPOSED SCHEMES WITH RECENT SELECTIVE ENCRYPTION TECHNIQUES FOR VIDEO CODECS

SE Scheme	C1	C2	C3	C4	C5	C6	C7
Yeung *et al.* [31]	No	No	Transform	Yes	No	Permutation	No
Jiang *et al.* [9]	No	No	IPM	Yes	No	Chaotic based PRNG	No
Dufaux and Ebrahimi [6]	Yes	No	Transform	Yes	Yes	Pseudo random sign inversion	No
Li *et al.* [15]	No	No	NAL	No	No	Stream cipher	No
Lian *et al.* [16]	No	No	Transform	No	No	Stream cipher	No
Carrillo *et al.* [4]	Yes	Yes	Pixel	Yes	Yes	Pseudo random pixel permutations	No
Shahid *et al.* (SE-CABAC) [22]	Yes	No	Binstrings	No	No	AES-CFB	No
Proposed scheme	**Yes**	**No**	**Binstrings**	**No**	**No**	**AES-CFB**	**Yes**

C1: Format compliance, C2: Robust to transcoding, C3: Encryption domain, C4: Bitrate increase,
C5: Compression independence, C6: Encryption algorithm, C7: Context modeling for binarization



(a) (b)

Fig. 10. Laplace edge detection for original and encrypted frame #0 of the *foreman* video sequence for QP value 18: a) original frame, b) SE-CABAC of H.264/AVC ([22, Fig. 9.a]). It is evident that the edges are not as strongly protected in [22] as in the proposed SE-HEVC scheme. (a) Original. (b) SE-CABAC (H.264/AVC).

shows the edge information in the original and encrypted frame # 0 of *foreman* sequence taken from [22]. It is evident that edges and contours are not as well protected as in Fig. 7 for SE-HEVC. Moreover, the EDR (edge differential ratio) for SE-CABAC of H.264/AVC is 0.8, which is much lower in comparison to 0.94 of SE-HEVC and verifies that the proposed scheme offers better protection by distortion of the structural information.

VI. CONCLUSION

The paper presents a pioneer scheme for format compliant visual protection of HEVC using selective encryption. It starts with an in-depth analysis of HEVC-CABAC from an encryption standpoint. It is followed by the proposed algorithm for SE of HEVC, which fulfills all the real-time constraints, including conversion of non-dyadic ES to dyadic ES. The SE is performed on the entropy slices independently in HEVC. In this way, the proposed SE method does not affect the parallelism of HEVC. Moreover, SE is performed in CABAC *binstrings* such that they remain valid *binstrings* thereafter having exactly the same length. The proposed method has the advantage of being suitable for streaming over heterogeneous networks because of no change in bit-rate. The experiments have shown that we can achieve the desired level of protection, both for texture information using SE of QTC syntax elements and motion information using SE of MVD syntax elements, under a minimal set of computational requirements.

REFERENCES

[1] H. Ahmed, H. Kalash, and O. Allah, "Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images," in *Proc. Int. Conf. Electrical Engineering*, Apr. 2007, pp. 1–7.

[2] M. Asghar, M. Ghanbari, and M. Reed, "Sufficient encryption with codewords and bin-strings of H.264/SVC," in *Proc. IEEE Int. Conf. Trust, Security and Privacy in Computers and Communications*, Liverpool, U.K., Jun. 2012, pp. 443–450.

[3] V. Baroncini, J.-R. Ohm, and G. Sullivan, "Report of Subjective Test Results of Responses to the Joint Call for Proposals (CfP) on Video Coding Technology for High Efficiency Video Coding (HEVC)," Tech. Rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, 2010, Doc. JCTVC-A204.

[4] P. Carrillo, H. Kalva, and S. Magliveras, "Compression independent reversible encryption for privacy in video surveillance," *EURASIP J. Inf. Security*, vol. 2009, p. 13, 2009.

[5] L. Dubois, W. Puech, and J. Blanc-Talon, "Fast protection of H.264/AVC by reduced selective encryption of CAVLC," in *Proc. Eur. Signal Processing Conf.*, Barcelona, Spain, 2011, pp. 2185–2189.

[6] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1168–1174, Aug. 2008.

[7] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 905–917, Oct. 2006.

[8] HEVC, "High Efficiency Video Coding (HEVC) Text Specification Draft 6," Tech. Rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, USA, 2012, Doc. JCTVC-H1003.

[9] M. Q. J. Jiang and S. Xing, "An intra prediction mode-based video encryption algorithm in H.264," in *Proc. Int. Conf. Multimedia Information Networking and Security*, Nov. 2009, vol. 1, pp. 478–482.

[10] G. Jakimoski and K. Subbalakshmi, "Cryptanalysis of some multimedia encryption schemes," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 330–338, Apr. 2008.

[11] W. Jiangtao, K. Hyungjin, and J. Villasenor, "Binary arithmetic coding with key-based interval splitting," *IEEE Signal Process. Lett.*, vol. 13, no. 2, pp. 69–72, Feb. 2006.

[12] C. Kim, J. Kim, and J. Park, "Simplification of Golomb-Rice Parameter Update," Tech. Rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, 2012, JCTVC-10124.

[13] J. Lainema and K. Ugur, "Angular intra prediction in high efficiency video coding (HEVC)," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, Hangzhou, China, Oct. 2011, pp. 1–5.

[14] H.-J. Lee and J. Nam, "Low complexity controllable scrambler/descrambler for H.264/AVC in compressed domain," in *Proc. ACM Int. Conf. Multimedia*, New York, NY, USA, 2006, pp. 93–96.

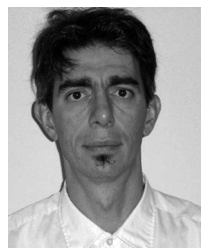
[15] C. Li, X. Zhou, and Y. Zong, "NAL level encryption for scalable video coding," *Lecture Notes Comput. Sci., Springer*, no. 5353, pp. 496–505, 2008.

[16] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 774–778, Jun. 2007.

[17] S. Lian, Z. Liu, Z. Ren, and Z. Wang, "Selective video encryption based on advanced video coding," *Lecture Notes Comput. Sci., Springer-Verlag*, no. 3768, pp. 281–290, 2005.

[18] T. Nguyen, "CE11: Coding of Transform Coefficient Levels with Golomb-Rice Codes," Tech. Rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, 2011, Doc. JCTVC-E253.
[19] S. Park and S. Shin, "An efficient encryption and key management scheme for layered access control of H.264/Scalable video coding," *IEICE Trans. Inf. Syst.*, vol. 92, no. 5, pp. 851–858, 2009.
[20] A. Said, "Measuring the strength of partial encryption scheme," in *Proc. IEEE Int. Conf. Image Processing*, Genoa, Italy, 2005, vol. 2, pp. 1126–1129.
[21] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CABAC for I & P frames," in *Proc. 17th Eur. Signal Process. Conf. (EUSIPCO'09)*, Glasgow, U.K., Aug. 2009, pp. 2201–2205.
[22] Z. Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I & P frame," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 565–576, May 2011.
[23] H. Sohn, E. AnzaKu, W. De-Neve, Y. Ro, and K. Plataniotis, "Privacy protection in video surveillance systems using scalable video coding," in *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance*, Genoa, Italy, Sep. 2009, pp. 424–429.
[24] N. Taneja, B. Raman, and I. Gupta, "Chaos based partial encryption of spiht compressed images," *Int. J. Wavelets, Multires., Inf. Process.*, vol. 9, no. 2, pp. 317–331, 2011.
[25] K. Ugur, K. Andersson, and A. Fuldseth, "Video Coding Technology Proposal by Tandberg, Nokia, and Ericsson," Tech. Rep., Joint Video Team (JVT), Dresden, Germany, 2010, JCTVC-A119.
[26] D. Wang, Y. Zhou, D. Zhao, and J. Mao, "A partial video encryption scheme for mobile handheld devices with low power consideration," in *Proc. Int. Conf. Multimedia Information Networking and Security*, Nov. 2009, vol. 2, pp. 99–104.
[27] J. Wang, Y. Fan, T. Ikenaga, and S. Goto, "A partial scramble scheme for H.264 video," in *Proc. 7th Int. Conf. ASIC*, Guìlín, China, Oct. 2007, pp. 802–805.
[28] M. Winken, P. Helle, D. Marpe, H. Schwarz, and T. Wiegand, "Transform coding in the HEVC test model," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2011, pp. 3693–3696.
[29] Y. Won, T. Bae, and Y. Ro, "Scalable protection and access control in full scalable video coding," in *Proc. Int. Conf. Digital Watermarking*, Berlin, Germany, 2006, pp. 407–421.
[30] C. Wu and C. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Trans. Multimedia*, vol. 7, pp. 828–839, Oct. 2005.
[31] S. K. A. Yeung, S. Zhu, and B. Zeng, "Design of new unitary transforms for perceptual video encryption," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1341–1345, Sep. 2011.
[32] S. K. A. Yeung, S. Zhu, and B. Zeng, "Perceptual video encryption using multiple 8×8 transforms in H.264 and MPEG-4," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, May 2011, pp. 2436–2439.

Zafar Shahid received his BS (Electrical Engineering) degree with honors from University of Engineering and Technology Lahore, Pakistan in 2001. After his B.S., he worked as senior embedded system engineer with Streaming Networks, where he was involved in research and development of real-time video codecs until 2006. He obtained his MS degree in image processing from National Institute of Applied Sciences (INSA) Lyon, France in 2007 and PhD from University of Montpellier II France in 2010. His current research interests include compression, watermarking, encryption and scalable video.



William Puech received the diploma of Electrical Engineering from the University of Montpellier, France, in 1991 and the Ph.D. Degree in Signal-Image-Speech from the Polytechnic National Institute of Grenoble, France in 1997. He started his research activities in image processing and computer vision. He served as a Visiting Research Associate to the University of Thessaloniki, Greece. From 1997 to 2000, he had been an Assistant Professor in the University of Toulon, France, with research interests including methods of active contours applied to medical images sequences. Between 2000 and 2008, he had been Associate Professor and since 2009, he is full Professor in image processing at the University of Montpellier, France. He works in the LIRMM Laboratory (Laboratory of Computer Science, Robotic and Microelectronic of Montpellier). His current interests are in the areas of protection of visual data (image, video and 3D object) for safe transfer by combining watermarking, data hiding, compression and cryptography. He has applications on medical images, cultural heritage and video surveillance. He is the head of the ICAR team (Image & Interaction) and he has published more than 15 journal papers, 8 book chapters and more than 80 conference papers. W. Puech is associate editor of J. of Advances in Signal Processing, Springer and he is reviewer for more than 15 journals (IEEE Trans. on Image Processing, IEEE Trans. on Multimedia, IEEE TCSVT, IEEE TIFS, Signal Processing: Image Communication, etc.) and for more than 10 conferences (IEEE ICIP, EUSIPCO, etc.).